REMARKS

In view of the following discussion, the Applicants submit that none of the claims now pending in the application are made obvious under the provisions of 35 U.S.C. §103. Thus, the Applicants believe that all of these claims are now in allowable form.

I. REJECTION OF CLAIMS 1-42 UNDER 35 U.S.C § 103

A. Claims 1, 4-5, 10-11, 14-15, 17, 20-23, 26-28, 31-32, 35-36 and 40-42

Claims 1, 4-5, 10-11, 14-15, 17, 20-23, 26-28, 31-32, 35-36 and 40-42 stand rejected as being made obvious by the Nepustil patent (U.S. Patent No. 6,240,454, issued May 29, 2001, hereinafter "Nepustil") in view of the Bhanot et al. patent (U.S. Patent No. 5,796,934, issued August 18, 1998, hereinafter "Bhanot"). The Applicants respectfully traverse the rejection.

Nepustil teaches a dynamic reconfiguration of network servers. Nepustil discloses a plurality of servers for processing client requests, wherein at least one first server of the plurality of servers has first information and second information related to the first information, for processing portions of the client requests that require the first information and portions of the client requests that require the second information. (See Nepustil, col. 2, ll. 20-46.) If processing on the at least one server becomes excessive, then the at least one server processes the portions of the client requests which require the first information without also processing the portions of the client requests which require the second information. (See *Id.*) The portions of the client request which require the second information is redirected to at least one second server for processing. (See *Id.*)

Bhanot teaches a fault tolerant client server system in which a primary server normally handles all of a client's transactions. A backup server is designated to handle the client's transactions in the event that the primary server is disabled. Thus, if the client determines that the primary server is disabled, all pending transactions on the disabled primary server are rolled back and resubmitted to the backup server, such that the backup server can complete any transactions which were in progress on the primary server at the time of failure. (See, Bhanot, Abstract).

The Examiner's attention is directed to the fact that Nepustil and Bhanot, singly or in any permissible combination, fail to teach, show or suggest determining a load on a primary server and offloading a processing request to any one of a plurality of offload servers only if a processing threshold is exceeded at the primary server, wherein the offloaded at least a portion of said processing requests is the only work handled by said plurality of offload servers, as positively claimed by the Applicants. Specifically, Applicants' independent claims 1, 11, 21, 22, 23, 32, 41 and 42 recite:

> 1. A method, in a network comprising a primary server and a plurality of offload servers, for dynamic offloading of processing requests from said primary server to any one of said plurality of offload servers, the method comprising the steps of:
>
> determining a load on said primary server;
>
> if the load on said primary server is less than a first threshold, serving processing requests at said primary server; and
>
> only if the load on said primary server exceeds said first threshold, then offloading at least a portion of said processing requests to any one of said plurality of offload servers, wherein all of said plurality of offload servers are configured to process said processing request and the offloaded at least a portion of said processing requests is the only work handled by said plurality of offload servers. (Emphasis added)

> 11. A network apparatus comprising a primary server and a plurality of offload servers connected by an IP-based network, for dynamic offloading of processing requests from said primary server to any one of said plurality of offload servers, the network apparatus comprising:
>
> a load controller connected between said network and said primary server;
>
> a memory connected to said load controller and including data and control instructions for operating said primary server to perform the steps of:
>
> determining a load on said primary server;
>
> if said load on said primary server is less than a first threshold, serving processing requests at said primary server; and
>
> only if said load on said primary server exceeds said first threshold, then offloading at least a portion of said processing requests to any one of said plurality of offload servers, wherein all of said plurality of offload servers are configured to process said processing request and the offloaded at least a portion of said processing requests is the only work handled by said plurality of offload servers. (Emphasis added)

21. A system, including an IP network comprising a primary server and a plurality of offload servers, for dynamic offloading of processing requests from said primary server to any one of said plurality of offload servers, the system comprising:

means for determining a load on said primary server;

means for, if said load on said primary server is less than a first threshold, serving the processing requests at said primary server; and

means for, only if said load on said primary server exceeds said first threshold, offloading at least a portion of said processing requests to any one of said plurality of offload servers, wherein all of said plurality of offload servers are configured to process said processing request and the offloaded at least a portion of said processing requests is the only work handled by said plurality of offload servers. (Emphasis added)

22. A program product including control instructions for controlling the operation of a computer, said program product operative with said control instructions to operate said computer in an IP-based network comprising a primary server and a plurality of offload servers to dynamically offload processing requests from said primary server to any one of said plurality of offload servers, the computer operative with said control instructions to perform the steps of:

determining a load on said primary server;

if said load on said primary server is less than a first threshold, serving the processing requests at said primary server; and

only if said load on said primary server exceeds said first threshold, then offloading at least a portion of said processing requests to any one of said plurality of offload servers, wherein all of said plurality of offload servers are configured to process said processing request and the offloaded at least a portion of said processing requests is the only work handled by said plurality of offload servers. (Emphasis added)

23. A system for allocating processing requirements on a network between a primary server and a plurality of offload servers, comprising:

a load controller connected to said network for receiving processing requests from clients on said network and allocating said processing requests between said primary and offload servers;

a memory connected to said load controller and storing threshold data and control software for analyzing said threshold data and operating said load controller;

said load controller operative with the threshold data and control software to perform the steps of:

periodically evaluating said processing requests to determine a load on said primary server;

if said load exceeds a first threshold, for a predetermined period of

time directing at least one processing request to any one of said plurality of offload servers, wherein all of said plurality of offload servers are configured to process said processing request and the offloaded at least a portion of said processing requests is the only work handled by said plurality of offload servers; and

only if said load does not exceed said first threshold, directing said processing requests to said primary server. (Emphasis added)

32. A method for allocating processing requirements on an IP network between a primary server and a plurality of offload servers, comprising:

periodically evaluating processing requests to determine a load on said primary server;

if said load exceeds a first threshold, for a predetermined period of time directing at least one processing request to any one of said plurality of offload servers, wherein all of said plurality of offload servers are configured to process said processing request and the offloaded at least a portion of said processing requests is the only work handled by said plurality of offload servers; and

only if said processing load does not exceed said first threshold, directing said processing requests to said primary server. (Emphasis added)

41. A system for allocating processing requirements on an IP network between a primary server and a plurality of offload servers, comprising:

means for periodically evaluating processing requests to determine a load on said primary server;

means for, if said load exceeds a first threshold, for a predetermined period of time directing at least one processing request to any one of said plurality of offload servers, wherein all of said plurality of offload servers are configured to process said processing request and the offloaded at least a portion of said processing requests is the only work handled by said plurality of offload servers; and

means for, only if said processing load does not exceed said first threshold, directing said processing requests to said primary server. (Emphasis added)

42. A program product including control instructions for controlling the operation of a computer, said program product operative with said control instructions to operate said computer in an IP-based network comprising a primary server and a plurality of offload servers to dynamically offload processing requests from said primary server to any one of said plurality of offload servers, the computer operative with said control instructions to perform the steps of:

periodically evaluating processing requests to determine a load on said primary server;

if said load exceeds a first threshold, for a predetermined period of time directing at least one processing request to any one of said plurality of offload servers, wherein all of said plurality of offload servers are configured to process said processing request and the offloaded at least a portion of said processing requests is the only work handled by said plurality of offload servers; and

only if said load does not exceed said first threshold, directing said processing requests to said primary server. (Emphasis added)

Applicants' invention is directed to a system and method for dynamically allocating processing on a network amongst multiple network servers. As the Internet continues to grow, so does traffic (*e.g.*, processing requests) directed to popular Internet servers on the World Wide Web. To support these high traffic rates, many techniques have been developed that use offload servers associated with primary web servers to process some of the traffic targeted to primary web servers. In many conventional cases, traffic is directed to these offload servers regardless of the current resource availability at the primary web servers. However, it is normally only during peak traffic periods that the offload servers are actually needed; during non-peak periods, substantial amounts of processing resources at the primary server may go unused due to the use of the offload servers. Thus, resources are wasted by using offload servers when they are not needed.

Applicants' invention provides a method for dynamically offloading traffic from a primary server to any one of a plurality of offload servers based on the current load at the primary server and one or more thresholds. In one particular embodiment, the load at the primary server is first determined. If the load falls below a first threshold (*e.g.*, the primary server is not currently over-loaded), then the traffic is directed to the primary server. However, if the load at the primary server exceeds the first threshold (*e.g.*, the primary server is over-loaded or is currently processing the maximum desirable amount of traffic), then at least a portion of the traffic is directed to any one of the plurality of offload servers. The first threshold may be based on a number of parameters, including network load, CPU utilization, connections per second, various bandwidth loads, various memory loads and the like. In this way, a web site can make use of excess processing capacity, offloading traffic only when the traffic exceeds the web site's processing capacity. Notably, the offloading servers are only used to handle the part of the load

that exceeds the primary server's capacity (*i.e.*, the excess load is the only work handled by the offloading servers). Thus, the Applicants' invention provides cost savings by drastically reducing offloaded work, while enabling the full load to be processed by a primary server operating simultaneously with one or more offloading servers.

The Examiner acknowledges that "Nepustil does not explicitly indicate that the offloaded server only handles s [sic] the portion of said processing requests" (Office Action, Page 3). The Examiner submits, however, that Bhanot bridges this gap in the teachings of Nepustil. The Applicants respectfully disagree.

Bhanot, by contrast, teaches that the full load of processing requests is handled by either a primary server or a backup server. That is, the backup server taught by Bhanot is not configured to handle only a portion of the load that exceeds a threshold load, but is configured to handle "all in-flight transactions" that were pending on the primary server at the time that the primary server failed (Bhanot, Abstract, emphasis added). Thus, the backup server actually takes on the role of the disabled primary server, processing all transactions that normally would have been processed by the primary server.

Moreover, Nepustil actually teaches away from combination with Bhanot. Whereas Nepustil specifically teaches that each server is both a primary server and an offload server (See, *e.g.*, Nepustil, col. 3, ll. 30-34), Bhanot teaches that a server is either a primary server or a backup server (but not both), and in either case the server (primary or backup) will handle the entire load when called in to operation (*See, e.g.*, Bhanot, Abstract).

Consequently, Nepustil in view of Bhanot fails to teach, show, or suggest the Applicants' invention and to render obvious independent claims 1, 11, 21, 22, 23, 32, 41 and 42. Furthermore, dependent claims 4-5, 10, 14-15, 17, 20, 26-28, 31, 35-36 and 40 depend, either directly or indirectly, from claims 1, 11, 21, 22, 23, and 32, and recite additional limitations. As such, and for at least the exact same reason set forth above, the Applicants submit that claims 4-5, 10, 14-15, 17, 20, 26-28, 31, 35-36 and 40 are also patentable and not made obvious by Nepustil in view of Bhanot. As such, the

Applicants respectfully request the rejection of claims 1, 4-5, 10-11, 14-15, 17, 20-23, 26-28, 31-32, 35-36 and 40-42 under 35 U.S.C. §103 be withdrawn.

2. Claims 2-3, 6-7, 12-13, 16, 24-25, 29, 33-34 and 37-39

Claims 2-3, 6-7, 12-13, 16, 24-25, 29, 33-34 and 37-39 stand rejected as being obvious over the Nepustil in view of Bhanot and further in view of the Swildens et al. patent (U.S. Patent No. 6,694,358, issued February 17, 2004, hereinafter "Swildens"). The Applicants respectfully traverse the rejection.

The teachings of Nepustil and Bhanot are discussed above. Swildens teaches a performance computer network method. Specifically, Swildens teaches a load balancing method that determines the traffic loads (*e.g.*, volume of processing requests) on a plurality of web servers. These various traffic loads are then compared to identify the web server that has the smallest traffic load among the plurality of web servers, and traffic is directed to this server.

The Examiner's attention is directed to the fact that Nepustil, Bhanot, and Swildens, singly and in any permissible combination, fail to teach, show or suggest determining a load on a primary server and offloading a processing request to any one of a plurality of offload servers only if a processing threshold is exceeded at the primary server, wherein the offloaded at least a portion of said processing requests is the only work handled by said plurality of offload servers, as positively claimed by the Applicants' independent claims 1, 11, 21, 22, 23, 32, 41 and 42. (See *supra*.)

As discussed above, Nepustil in view of Bhanot fails to teach Applicants' invention. Swildens fails to bridge this gap in the teachings of Nepustil and Bhanot. Specifically, Swildens fails to teach or to suggest a preference for the use of a specified or primary server (*e.g.*, to reduce offloading costs). Rather Swildens requires the monitoring of a plurality of servers that may potentially be used to process traffic. Thus, Swildens clearly teaches away from the Applicants' invention because Swildens requires more data collection than the Applicants' invention does. Applicants' invention determines the load on only one server (*e.g.*, the primary server), unlike Swildens, which determines the load on a plurality of servers. Because it monitors fewer servers,

the Applicants' invention requires less calculations and less comparisons and therefore consumes less time than systems that determine the load traffic data at a plurality of servers, compare the loads of each server to the loads of other servers and/or respective thresholds and determine which web server to use for processing from among the plurality (*e.g.*, as taught by Swildens).

Moreover, the Applicants respectfully submit that Nepustil, Bhanot, and Swildens cannot be meaningfully combined because Swildens teaches away from Nepustil. Nepustil teaches specific offload server assignment, as discussed above. Bhanot teaches a specific backup server to be used in the event that a primary server fais. In contrast, Swildens teaches that an optimal sever is chosen from among a plurality of possible servers based on various factors. (See Swildens, col. 2, ll. 46-64). Thus, the choice of which server to allocate a load to, as taught by Swildens, is not predetermined (*e.g.*, according to a predefined assignment).

Therefore, Applicants respectfully submit that independent claims 1, 11, 21, 22, 23, 32, 41 and 42 are clearly patentable and not made obvious by Nepustil in view of Bhanot and further in view of Swildens. Furthermore, dependent claims 2-3, 6-7, 12-13, 16, 24-25, 29, 33-34 and 37-39 depend, either directly or indirectly, from claims 1, 11, 21, 22, 23, and 32 and recite additional limitations. As such, and for at least the exact same reason set forth above, the Applicants submit that claims 2-3, 6-7, 12-13, 16, 24-25, 29, 33-34 and 37-39 are also patentable and not made obvious by Nepustil in view of Bhanot and further in view of Swildens. As such, the Applicants respectfully request the rejection of claims 2-3, 6-7, 12-13, 16, 24-25, 29, 33-34 and 37-39 under 35 U.S.C. § 103 be withdrawn.

3. Claims 8-9, 18-19 and 30

Claims 8-9, 18-19 and 30 stand rejected as being obvious over Nepustil, Bhanot, and Swildens in view of the Gupta et al. patent (U.S. Patent No. 6,374,305, issued April 16, 2002, hereinafter "Gupta"). The Applicants respectfully traverse the rejection.

The teachings of Nepustil, Bhanot, and Swildens have been discussed above. Gupta teaches a web applications interface system in a mobile-based client-server

system. Specifically, Gupta teaches architecture that incorporates two specialized software layers: a specialized "proxy" layer that resides on a mobile client station, and a "web agent" that resides on a server. These layers employ respective memory caches and intelligent filtering capabilities, thereby reducing redundant or otherwise unwanted message transmission.

As discussed above, Nepustil, Bhanot, and Swildens fail to teach, show or suggest determining a load on a primary server and offloading a processing request to any one of a plurality of offload servers only if a processing threshold is exceeded at the primary server, wherein the offloaded at least a portion of said processing requests is the only work handled by said plurality of offload servers, as positively claimed by the Applicants' independent claims 1, 11 and 23. Gupta fails to bridge this gap in the teachings of Nepustil, Bhanot, and Swildens.

Thus, claims 1, 11 and 23 are not made obvious by Nepustil, Bhanot, and Swildens in view of Gupta. Dependent claims 8-9, 18-19 and 30 depend, either directly or indirectly, from claims 1, 11 and 23 and recite additional limitations. As such, and for at least the exact same reasons set forth above, the Applicants submit that claims 8-9, 18-19 and 30 are also not made obvious by the teachings of Nepustil, Bhanot, and Swildens in view of Gupta.

III. CONCLUSION

Thus, the Applicants submit that all of the presented claims fully satisfy the requirements of 35 U.S.C. §103. Consequently, the Applicants believe that all of the presented claims are presently in condition for allowance. Accordingly, both reconsideration of this application and its swift passage to issue are earnestly solicited.

If, however, the Examiner believes that there are any unresolved issues requiring the maintenance of the present final action in any of the claims now pending in the application, it is requested that the Examiner telephone Mr. Kin-Wah Tong, Esq. at (732) 530-9404 so that appropriate arrangements can be made for resolving such issues as expeditiously as possible.

PATENT
Atty. Dkt. No. YOR920010320US1

Respectfully submitted,

January 7, 2008
Date

Kin-Wah Tong, Attorney
Reg. No. 39,400
(732) 530-9404

Patterson & Sheridan, LLP
595 Shrewsbury Avenue
Shrewsbury, New Jersey 07702